

## **Тематика**

1. Олімпіадна інформатика
2. Тематика завдань
3. Системи тестування (ejudge, e-olymp, Contester )
4. Оптимізація розв'язку (прості числа, лічилка)
5. Довга арифметика (факторіал, фібоначі)
6. Повний перебір (лексичний перебір)
7. Обчислювальна геометрія (ліві повороти, площа многокутника)
8. Теорія графів (пошук в глибину, остове дерево, алгоритм флойда)
9. Практикум з розв'язування задач

## **1.Олімпіадна інформатика**

Що потрібне для успішної участі в олімпіадах по інформатиці? Як показує практика, тільки знання мови програмування для цього явно не достатньо. Взагалі, виявляється, що олімпіадні задачі по інформатиці лежать десь на стику математики і програмування. І дуже часто виявляється, що розв'язуючи задачі школярі не тільки вчаться програмувати, але і освоюють нові розділи математики.

### **Як перевіряються розв'язки задач олімпіади**

Історично склалося так, що розв'язки на олімпіадах по програмуванню перевіряються автоматично. Ваше завдання написати програму, яка за заданими вхідними даними обчислює і виводить вихідні дані.

### **Апаратне та програмне забезпечення**

Учасникам олімпіади можуть вибирати мову програмування з заданого переліку: Pascal, C або C++. Система програмування Free Pascal 2.0 (чи новішої версії), GCC 4.2 (чи новішої версії), Turbo Delphi Explorer, Visual C++ 2008 Express).

## **2. Тематика задача олімпіади**

Завдання олімпіади мають бути алгоритмічного характеру, тобто основними результатами роботи учасника має бути: алгоритм, що правильно та ефективно розв'язує поставлену задачу, та програма, що реалізує запропонований алгоритм.

**Основними категоріями олімпіадних задач є:**

Геометрія

Графічні задачі

Динамічне програмування

Довга арифметика

Жадібний алгоритм

Задачі для початківців

Комбінаторика

Масиви

Математика

Математичне моделювання

Обробка рядків

Послідовності

Рекурсія, перебір

Логічні задачі

Сортування

Структури даних

Теорія графів

Теорія ігор

Теорія чисел

### **3.Системи тестування (ejudge, e-olymp, Contester )**

Працюйте в он-лайн системах, які автоматично перевіряють та тестують Ваші розв'язки:

<http://olymp.sumdu.edu.ua> - Веб-ресурс підтримки та проведення шкільних та студентських олімпіад з інформатики

<http://www.e-olimp.com.ua/> - Інтернет-портал організаційно-методичного забезпечення дистанційних олімпіад з програмування для обдарованої молоді навчальних закладів України

<http://www.olymp.vinnica.ua/> - Центр підтримки та проведення олімпіад школярів з використанням можливостей Internet.

### Список тестуючи систем

| Назва |
|-------|
|-------|

| Сайт |
|------|
|------|

| Система |
|---------|
|---------|

|        |
|--------|
| ejudge |
|--------|

[ejudge.ru](http://ejudge.ru)

|       |
|-------|
| Linux |
|-------|

Contester

[contester.ru](http://contester.ru)

Windows, Linux

### **Список ресурсів**

<http://vipplabinfo.16mb.com> – сайт «Лабораторія інформатики сьогодні», методична підтримки напрямків роботи.

<http://vipoolimp.16mb.com> – Волинська учнівська Інтернет олімпіада з програмування.

<http://schoololymp.byethost32.com> – заочна школа роботи з обдарованими учнями з інформатики.

### **4. Оптимізація розв'язку (прості числа, лічилка)**

За означенням поодільності на 1 та саме себе

Решето Ератосфена

## Методика підготовки до олімпіади (навчаємось постійно)

Добавил(а) Гісь Ігор Володимирович

11.06.13 11:19 - Последнее обновление 11.06.13 11:38

---

program prog1;

{\$APPTYPE CONSOLE}

uses

SysUtils,

windows;

var n,i,j:integer;

p:integer;

time:int64;

begin

readln(n);

## Методика підготовки до олімпіади (навчаємось постійно)

Добавил(а) Гісь Ігор Володимирович

11.06.13 11:19 - Последнее обновление 11.06.13 11:38

---

```
time:=gettickcount;
```

```
for i:=2 to n do
```

```
begin
```

```
p:=0;
```

```
for j:=2 to round(sqrt(i)) do
```

```
if i mod j =0 then p:=1;
```

```
if p=0 then write(i, ' ');
```

```
end;
```

```
writeln;
```

```
time:=gettickcount-time;
```

```
writeln(time/1000:0:5);
```



## Методика підготовки до олімпіади (навчаємось постійно)

Добавил(а) Гісь Ігор Володимирович

11.06.13 11:19 - Последнее обновление 11.06.13 11:38

---

readln;

end.

program prog2;

{\$APPTYPE CONSOLE}

uses

SysUtils,

windows;

var a:array[1..100000000] of integer;

j,k,n,i:integer;

time:int64;

begin

## Методика підготовки до олімпіади (навчаємось постійно)

Добавил(а) Гісь Ігор Володимирович

11.06.13 11:19 - Последнее обновление 11.06.13 11:38

---

```
readln(n);
```

```
time:=gettickcount;
```

```
for i:=1 to n do a[i]:=i;
```

```
a[1]:=0;
```

```
i:=1;
```

```
while i<=n div 2 do begin
```

```
while a[i]=0 do i:=i+1;
```

```
//writeln(i);readln;
```

```
j:=i+a[i];
```

```
while j<=n do begin
```

```
a[j]:=0; j:=j+a[i];
```

```
end;
```

## Методика підготовки до олімпіади (навчаємось постійно)

Добавил(а) Гісь Ігор Володимирович

11.06.13 11:19 - Последнее обновление 11.06.13 11:38

---

i:=i+1;

//for k:=1 to n do write(a[k], ' ');readln;

end;

for k:=1 to n do

if a[k]<>0 then ///write(a[k], ' ');

writeln;

time:=gettickcount-time;

writeln(time/1000:0:5);

readln;

end.

## 5.Довга арифметика (факторіал, фібоначі)

Цикл

## Методика підготовки до олімпіади (навчаємось постійно)

Добавил(а) Гісь Ігор Володимирович

11.06.13 11:19 - Последнее обновление 11.06.13 11:38

---

|          |
|----------|
| Рекурсія |
|----------|

|                  |
|------------------|
| Довгий факторіал |
|------------------|

|          |            |
|----------|------------|
| #include | "stdafx.h" |
|----------|------------|

|          |            |
|----------|------------|
| #include | "iostream" |
|----------|------------|

|       |           |      |
|-------|-----------|------|
| using | namespace | std; |
|-------|-----------|------|

|     |        |
|-----|--------|
| int | main() |
|-----|--------|

|   |     |      |
|---|-----|------|
| { | int | n,i; |
|---|-----|------|

|      |      |      |
|------|------|------|
| long | long | f=1; |
|------|------|------|

|         |
|---------|
| cin>>n; |
|---------|

|     |                |       |
|-----|----------------|-------|
| for | (i=1;i<=n;i++) | f*=i; |
|-----|----------------|-------|

|                |
|----------------|
| cout<<f<<endl; |
|----------------|

## Методика підготовки до олімпіади (навчаємось постійно)

Добавил(а) Гісь Ігор Володимирович

11.06.13 11:19 - Последнее обновление 11.06.13 11:38

---

|        |    |
|--------|----|
| return | 0; |
|--------|----|

|   |
|---|
| } |
|---|

|          |            |
|----------|------------|
| #include | "stdafx.h" |
|----------|------------|

|          |            |
|----------|------------|
| #include | "iostream" |
|----------|------------|

|       |           |      |
|-------|-----------|------|
| using | namespace | std; |
|-------|-----------|------|

|      |      |    |     |
|------|------|----|-----|
| long | long | f( | int |
|------|------|----|-----|

|   |
|---|
| { |
|---|

|      |      |    |
|------|------|----|
| long | long | t; |
|------|------|----|

|    |             |
|----|-------------|
| if | (i==0) t=1; |
|----|-------------|

## Методика підготовки до олімпіади (навчаємось постійно)

Добавил(а) Гісь Ігор Володимирович

11.06.13 11:19 - Последнее обновление 11.06.13 11:38

---

|      |             |
|------|-------------|
| else | t=i*f(i-1); |
|------|-------------|

|        |    |
|--------|----|
| return | t; |
|--------|----|

|   |
|---|
| } |
|---|

|     |        |
|-----|--------|
| int | main() |
|-----|--------|

|   |     |    |
|---|-----|----|
| { | int | n; |
|---|-----|----|

|         |
|---------|
| cin>>n; |
|---------|

|                   |
|-------------------|
| cout<<f(n)<<endl; |
|-------------------|

|        |    |
|--------|----|
| return | 0; |
|--------|----|

|   |
|---|
| } |
|---|

## Методика підготовки до олімпіади (навчаємось постійно)

Добавил(а) Гісь Ігор Володимирович

11.06.13 11:19 - Последнее обновление 11.06.13 11:38

---

|          |            |
|----------|------------|
| #include | "stdafx.h" |
|----------|------------|

|          |            |
|----------|------------|
| #include | "iostream" |
|----------|------------|

|       |           |      |
|-------|-----------|------|
| using | namespace | std; |
|-------|-----------|------|

|     |         |
|-----|---------|
| int | main () |
|-----|---------|

|   |
|---|
| { |
|---|

|     |           |
|-----|-----------|
| int | n,i,k,os; |
|-----|-----------|

|         |
|---------|
| cin>>n; |
|---------|

|     |                  |
|-----|------------------|
| int | a[1000],b[1000]; |
|-----|------------------|

|     |                         |
|-----|-------------------------|
| for | (i=0;i<1000;i++)a[i]=0; |
|-----|-------------------------|

|     |                         |
|-----|-------------------------|
| for | (i=0;i<1000;i++)b[i]=0; |
|-----|-------------------------|

|                       |
|-----------------------|
| a[0]=1;a[1]=1;b[0]=1; |
|-----------------------|

## Методика підготовки до олімпіади (навчаємось постійно)

Добавил(а) Гісь Ігор Володимирович

11.06.13 11:19 - Последнее обновление 11.06.13 11:38

---

|     |                |
|-----|----------------|
| for | (k=1;k<=n;k++) |
|-----|----------------|

|         |
|---------|
| { os=0; |
|---------|

|     |                    |
|-----|--------------------|
| for | ( i=1;i<=a[0];i++) |
|-----|--------------------|

|                       |
|-----------------------|
| {b[i]=(a[i]*k+os)%10; |
|-----------------------|

|                     |
|---------------------|
| os=(a[i]*k+os)/10;} |
|---------------------|

|       |                                        |
|-------|----------------------------------------|
| while | (os>0){b[0]++;b[b[0]]=os%10;os=os/10;} |
|-------|----------------------------------------|

|     |                              |
|-----|------------------------------|
| for | ( i=0;i<=b[0];i++)a[i]=b[i]; |
|-----|------------------------------|

|   |
|---|
| } |
|---|

|     |                   |
|-----|-------------------|
| for | (i=b[0];i>=1;i--) |
|-----|-------------------|

|             |
|-------------|
| cout<<b[i]; |
|-------------|

|        |      |   |
|--------|------|---|
| cout<< | "\n" | ; |
|--------|------|---|



## 6. Повний перебір (лексичний перебір)

Побудуємо лексичний перебір для довільних елементів масиву

$X = 3\ 2\ 4\ 2\ 4\ 3\ 1$

а) Рухаємось справа наліво. Крок вперед можна зробити, якщо наступне число більше за попереднє. Ми зупинилися перед числом 2. Це число потрібно помітити.

$X = 3\ 2\ 4\ \mathbf{2}\ 4\ 3\ 1$

б) Рухаємось справа наліво. Крок вперед можна зробити, якщо число менше за знайдене число(2). Ми зупинилися перед числом 3. Це число потрібно помітити.

$X = 3\ 2\ 4\ \mathbf{2}\ 4\ \mathbf{3}\ 1$

в) Переставляємо знайдені числа.

$X = 3\ 2\ 4\ \mathbf{3}\ 4\ \mathbf{2}\ 1$

г) Запишемо числа, розміщені після першого знайденого в зворотному порядку.

$X = 3\ 2\ 4\ \mathbf{3}\ 1\ \mathbf{2}\ 4$

## 7.Обчислювальна геометрія (ліві повороти, площа многокутника)

Ліві повороти

```
{Визначення координат векторів}
x[n+1]:=x[1];
y[n+1]:=y[1];
for i:=1 to n do begin
a[i]:=x[i+1]-x[i];
b[i]:=y[i+1]-y[i];
end;
{Підрахунок кількості додатних добутків}
a[n+1]:=a[1];
b[n+1]:=b[1];
k:=0;
for i:=1 to n do
if a[i]*b[i+1]-a[i+1]*b[i]>=0 then k:=k+1;
```

За заданими кількістю вершин і координатами n-кутника визначити його площу (15 балів).

$x[n+1]=x[1]$

$y[n+1]=y[1]$

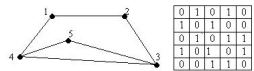
$s=0$

для  $i=1$  до  $n$

$s = s + (x[i] * y[i+1] - x[i+1] * y[i])$

$s = 1/2 * \text{abs}(s);$

### 8.Теорія графів (пошук в глибину, остове дерево, алгоритм флойда)



#### Основні поняття теорії графів

1. Граф представляє собою не порожню множину точок і ліній, два кінці котрих належать заданій множині точок.

Неорієнтований граф:

Відкриті Ейлером властивості графа:

1. Число непарних вершин зв'язного графа завжди парне. Неможливо накреслити граф з непарним числом непарних вершин.

2. Якщо всі вершини графа парні, то можна одним розчерком (тобто не відриваючи олівця від паперу) накреслити граф, не проводячи по кожному ребру більше одного разу. При цьому можна починати з будь-якої вершини графа і закінчувати його в тій же вершині.

3. Граф лише з двома непарними вершинами можна накреслити одним розчерком, при цьому рух потрібно почати в одній точці з цих вершин і закінчити в другій непарній вершині.

4. Граф з більше ніж двома непарними вершинами неможливо накреслити одним розчерком.

**Гамільтонів шлях** проходить через кожную вершину по одному разу (по ребрах проходить декілька разів або жодного)

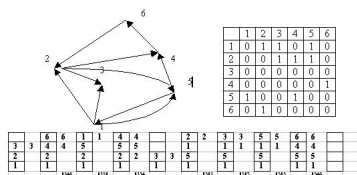
Ейлеровий шлях – це шлях, який ми проходимо з однієї вершини в іншу через всі ребра тільки один раз.

### Задача про туриста (пошук в глибину)

Турбаза мала для ночівлі  $N$  місць, з'єднаних стежками. Туристів можна вести в одну сторону. Довжина стежки – одноденний перехід. Пройти і перевірити всі  $M$ -денні маршрути, які починаються на базі  $K$ .

Орієнтований ненавантажений граф.

$n=6$ ;  $m=3$ ;  $k=1$ ;



Дано орієнтований ненавантажений граф. N вершин. Знайти всі маршрути довжини  $m=3$ , які виходять з вершини з номером k.

### Алгоритм Флойда-Уоршелла

З алгоритму Дейкстри зрозуміло, що можна знайти найкоротшу відстань від заданої вершини графа до решти його вершин. А якщо ця інформація потрібна для будь-якої вершини графа? Найперша відповідь, яка спадає на думку: необхідно виконати алгоритм Дейкстри в циклі для всіх вершин графа. Питання лише в часі виконання такого алгоритму, оскільки його оцінка буде  $O(n^3)$ . Однак існує компактніший за записом алгоритм Флойда-Уоршелла, з яким ми зараз і ознайомимось.

Запишемо сам алгоритм:

1. Визначити вершину графа  $k = 1$ , через яку буде здійснюватися перерахунок відстані між вершинами  $i$  та  $j$ .
2. Визначити вершину  $i = 1$ . Визначити вершину  $j = 1$ ,
3. Якщо величина  $d_{j,k} + d_{k,i}$  менша за значення  $d_{i,j}$ , то замінити значення  $d_{i,j}$  на  $d_{i,k} + d_{k,i}$ . В іншому разі залишити значення  $d_{i,j}$ .

$i, j$   
без змін.

4. Якщо  $j \leq n$ , то перейти до наступної вершини  $j + 1$  і повернутися до п. 4.

5. Якщо  $i \leq n$ , то перейти до наступної вершини  $i + 1$  і повернутися до п. 3.

6. Якщо  $k \leq n$ , то перейти до наступної вершини  $k + 1$  і повернутися до п. 2,

7. Завершити алгоритм.

У результаті виконання алгоритму елементи  $d_{i,j}$  будуть містити найкоротшу відстань між відповідними вершинами графа  $i$  та  $j$ .

Реалізація алгоритму Флойда-Уоршелла мовою Pascal буде такою:

for k := 1 to n do

for i := 1 to n do

for j := 1 to n do

if  $d[i,k] + d[k,j] < d[i,j]$  then  $d[i,j] := d[i,k] + d[k,j]$ ;

## Методика підготовки до олімпіади (навчаємось постійно)

Добавил(а) Гісь Ігор Володимирович

11.06.13 11:19 - Последнее обновление 11.06.13 11:38

---

4

0 20 3 2

20 0 40 3

3 40 0 4

2 3 4 0

0 5 3 2

5 0 7 3

3 7 0 4

2 3 4 0

0 20 3 2

## Методика підготовки до олімпіади (навчаємось постійно)

Добавил(а) Гісь Ігор Володимирович

11.06.13 11:19 - Последнее обновление 11.06.13 11:38

---

20 0 40 3

3 40 0 4

2 3 4 0

0 20 3 2

20 0 23 3

3 40 0 4

2 3 4 0

0 20 3 2

20 0 23 3

3 23 0 4

2 3 4 0

0 5 3 2



## Методика підготовки до олімпіади (навчаємось постійно)

Добавил(а) Гісь Ігор Володимирович

11.06.13 11:19 - Последнее обновление 11.06.13 11:38

---

20 0 23 3

3 23 0 4

2 3 4 0

0 5 3 2

5 0 23 3

3 23 0 4

2 3 4 0

0 5 3 2

5 0 7 3

3 23 0 4

2 3 4 0

0 5 3 2

5 0 7 3

3 7 0 4

2 3 4 0

## 9. Практикум з розв'язування задач

[http://nvk26.lutsk.ua/cgi-bin/new-client?contest\\_id=3](http://nvk26.lutsk.ua/cgi-bin/new-client?contest_id=3) – II етап (районний)

Всеукраїнської олімпіади з інформатики 2012-2013 н.р. (

логін user400-user430; пароль 400-430

).

[http://nvk26.lutsk.ua/cgi-bin/new-client?contest\\_id=11](http://nvk26.lutsk.ua/cgi-bin/new-client?contest_id=11) – III етап (обласний)

Всеукраїнської олімпіади з інформатики 2012-2013 н.р. (

логін user400-user430; пароль 400-430

).

[http://nvk26.lutsk.ua/cgi-bin/new-client?contest\\_id=12](http://nvk26.lutsk.ua/cgi-bin/new-client?contest_id=12) – тренувальний тур по

підготовці до IV етапу Всеукраїнської олімпіади з інформатики 2012-2013 н.р. (

логін user400-user414; пароль 400-414

).